

# **Estudi de maniobres per evitar obstacles en realitat virtual col·laborativa i immersiva**

---

**Treball de fi de grau realitzat a la  
Facultat d'Informàtica de Barcelona (FIB)  
Universitat Politècnica de Catalunya (UPC) – BarcelonaTech  
per  
Marc Palomar Soler**



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH**

---

**Grau d'Enginyeria Informàtica  
Computació**

**Director: Alejandro Ríos Jerez - Ciències de la Computació  
Codirectora: Nuria Pelechano Gomez - Ciències de la Computació  
Barcelona, Juny 2018**



## Resum

En aquest projecte es proposa desenvolupar un experiment en Unity 3D el qual ens proporcionï un context on estudiar el comportament de les maniobres dels usuaris a l'hora d'evitar obstacles mentre executen una tasca definida, en un entorn de realitat virtual. L'aplicació també disposarà d'un sistema d'enregistrament de les dades de l'usuari en el temps, les quals podran ser carregades i visualitzades amb el visualitzador, també programat en Unity 3D.

La finalitat d'aquest projecte serà proporcionar una aplicació per poder realitzar estudis que proporcionin dades per millorar el grau d'immersió dels usuaris en entorns de realitat virtual.

## Abstract

In this project we propose to develop an experiment in Unity 3D which provides us with a context in which to study the behavior of users' maneuvers when it comes to avoiding obstacles while performing a defined task in a virtual reality environment. The application will also have a system for recording user data over time, which can be loaded and displayed with the viewer, also programmed in Unity 3D.

The purpose of this project will be to propose an application to be able to carry out studies that provide data to improve the degree of immersion of users in virtual reality environments.

# Índex

<b>Resum</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>1. Formulació del problema</b>	<b>7</b>
<b>2. Context</b>	<b>7</b>
2.1. Contextualització	7
2.2. Objectius	8
<b>3. Actors implicats</b>	<b>8</b>
3.1. Desenvolupador	8
3.2. Tutors	8
3.3. Usuaris	8
<b>4. Estat de l'art</b>	<b>9</b>
4.1. Col·lisions entre usuaris en un entorn de realitat virtual	9
4.2. Avisar l'usuari amb un senyal acústic	9
4.3. Avisar l'usuari amb un senyal gràfica	9
4.4. Visibilitat dels avatars	9
<b>5. Abast</b>	<b>10</b>
5.1. Possibles obstacles	10
5.2. Programació de realitat virtual	10
5.3. Accessibilitat	11
<b>6. Metodologia i rigor</b>	<b>11</b>
6.1. Eines de desenvolupament	11
6.2. Eines de seguiment	11
6.3. Mètodes de validació	11
<b>7. Planificació</b>	<b>12</b>
7.1 Descripció de les tasques	12
7.1.1. Definició del projecte (duració aproximada: 7 dies)	12
7.1.2. Preparació de l'entorn (duració aproximada 14 dies)	12
7.1.3. Desenvolupament de l'escena (duració aproximada: 28 dies)	13
7.1.4. Experimentació amb usuaris i recollida de dades (duració aproximada: 14 dies)	13
7.1.5. Tractament de dades (duració aproximada: 14 dies)	13
7.1.6. Redacció de la memòria, l'annex i la documentació (duració aproximada: 21 dies)	14
<b>8. Duració total</b>	<b>14</b>
<b>9. Valoració d'alternatives</b>	<b>14</b>
<b>10. Diagrama de Gantt</b>	<b>15</b>

<b>11. Autoavaluació del domini actual de la competència de sostenibilitat</b>	<b>16</b>
<b>12. Dimensió Econòmica de la matriu de Sostenibilitat: Pressupost</b>	<b>17</b>
12.1. Pressupost de recursos humans	17
12.2. Pressupost de hardware	17
12.3. Pressupost de software	18
12.4. Despeses indirectes	18
12.5. Control de gestió	18
12.6. Cost total	19
<b>13. Dimensió Econòmica de la matriu de Sostenibilitat: Reflexió</b>	<b>20</b>
<b>14. Dimensió Ambiental de la matriu de Sostenibilitat</b>	<b>20</b>
<b>15. Dimensió Social de la matriu de Sostenibilitat</b>	<b>21</b>
<b>16. Desenvolupament</b>	<b>22</b>
16.1. Introducció	22
16.2. Configuració de l'entorn	22
12.2.1. Creació de l'habitació en Unity	23
12.2.2. Entorn online	23
12.2.3. Menús	24
12.2.4. Definició de l'entorn del hardware HTC Vive	26
12.2.5. Configuració de l'entorn del hardware HTC Vive	26
12.2.5.1. Steam VR	26
12.2.5.2. VRTK - Virtual Reality Toolkit	28
16.3. Configuració de l'experiment	28
16.3.1. Disseny de l'habitació	29
16.3.2. Configuració dels avatars	30
16.3.2.1. Posicionament dels avatars	30
16.3.2.2. Escalat dels avatars	30
16.3.2.3. Animació dels avatars	30
16.3.3. Configuració dels trencaclosques	31
16.3.3.1. Cos no kinematic	31
16.3.3.2. Reset del trencaclosques	32
16.3.3.3. Col·locar peça	32
16.3.4. Configuració de l'interacció en xarxa	33
16.3.4.1. Configuració de l'avatar en la xarxa	33
16.3.4.2. Funcions RPC (Remote Procedure Call)	34
16.3.4. Posicionament de l'escena	35
16.4. Recol·lecció i visualitzador de dades	36
16.4.1. Recol·lecció de dades	36
16.4.2. Visualitzador de dades	36
16.5. Realització de l'experiment	37

16.5.1. Preparació	37
16.5.2. Execució	38
16.5.3. Treball posterior	39
<b>17. Conclusions</b>	<b>39</b>
<b>18. Treball futur</b>	<b>40</b>
<b>Referències</b>	<b>41</b>
<b>Apèndix</b>	<b>42</b>
Posició, rotació i escalat de l'avatar	42
Animació de l'avatar	43
Col·locació de les peces	44
Captura i guardat de l'informació del usuari	45

# 1. Formulació del problema

Els cascos de realitat virtual s'han popularitzat molt en els últims anys, degut principalment al fet que els preus s'han reduït dràsticament (per exemple Oculus Rift 650 euros i HTC VIVE 700 euros) això ha permès que la realitat virtual deixi de ser utilitzada únicament en centres d'investigació i ha passat a convertir-se en una eina de treball en molts àmbits (per exemple arquitectura, jocs, medicina, psicologia, immobiliàries, etc.).

En aquest context sorgeix també la necessitat d'utilitzar la realitat virtual immersiva de forma col·laborativa, per exemple, diversos arquitectes volen observar un edifici que estan dissenyant de manera que puguin discutir diferents decisions, apuntar a llocs concrets i que l'altra persona ho vegi, modificar de forma interactiva el model, etc. Per tant, això obre un camp nou de treball en què cal estudiar noves tècniques d'interacció, així com noves metàfores de col·laboració que facilitin el treball dels experts. Un dels principals problemes que s'observen quan diverses persones col·laboren sol ser la falta de confiança en què el posicionament virtual dels altres usuaris sigui correcte i per tant no hi hagi perill de xocar físicament.

La conseqüència immediata d'aquest problema, és que les persones tendeixen a comportar-se i a moure de forma diferent en RV (realitat virtual) a com ho fan en el món real. Donada la novetat d'aquest tema, cal fer estudis del comportament humà en RV immersiva durant tasques que requereixen col·laboració.

## 2. Context

### 2.1. Contextualització

Crear un escenari amb dos avatars virtuals que es mouran seguint les trajectòries de dos usuaris immersos en el món virtual mitjançant cascos de realitat virtual. En aquest entorn volem estudiar el comportament en l'àmbit de moviment dels usuaris evitant xocar entre ells durant una tasca col·laborativa que impliqui caminar per un entorn de petites dimensions. Es pretén estudiar diverses configuracions de l'entorn: la primera per al cas en què l'animació dels avatars i so de les petjades va sincronitzat amb el moviment de l'usuari, el segon cas pel cas en que només l'animació va sincronitzada amb l'usuari, i finalment l'últim cas en que ni el so ni l'animació van sincronitzats.



## 2.2. Objectius

Per tal d'estudiar i resoldre aquests problemes en la interacció entre usuaris en un entorn de realitat virtual, marcarem els següents objectius:

1. Implementar els experiments que ens mostrin els episodis desitjats a estudiar. En aquest cas s'haurà d'implementar un experiment amb tres modes diferents de sincronització entre avatar i usuari.
2. Recol·lectar dades dels experiments utilitzant com a indicadors la proximitat dels avatars a l'hora de dur a terme una tasca concreta.
3. Tractament de les dades i conclusions.

## 3. Actors implicats

Els agents que participaran en aquest projecte són els següents.

### 3.1. Desenvolupador

Com a desenvolupador serè jo mateix (Marc Palomar Soler). M'encarregaré de programar amb Unity tots els experiments, preparar la interfície dels usuaris i preparar la recol·lecció de dades.

### 3.2. Tutors

L'Alejandro Rios i la Nuria Pelechano seràn els meus directors del projecte. En concret l'Alejandro Rios també serà el meu tutor i s'encarregarà de supervisar la meva feina, guiar-me i instruir-me i col·laborar a l'hora de dissenyar els experiments.

### 3.3. Usuaris

Durant els experiments hi participaran dues persones simultàniament, les quals es submergiran en el nostre entorn i seguiran les nostres indicacions.

## 4. Estat de l'art

### 4.1. Col·lisions entre usuaris en un entorn de realitat virtual

Actualment existeixen moltes aplicacions on és necessari compartir espai físic i/o virtual dins d'un entorn virtual entre usuaris, i evitar col·lisions entre ells ha estat estudiat en nombrosos estudis on s'ha arribat a proposar diferents solucions al problema.

### 4.2. Avisar l'usuari amb un senyal acústic

Blom & Beckhaus, en la investigació de la influència de la resposta en cas de col·lisió i la percepció de la realitat virtual [1], mostren que reproduir un senyal que indica col·lisió, acompanyada d'una resposta adequada amb el context, influeix positivament en el realisme de percepció de l'entorn virtual. En l'estudi, reproduint un so en cas de pròxima col·lisió amb un obstacle, els usuaris mostren un menor nombre de col·lisions durant l'experiment.

### 4.3. Avisar l'usuari amb un senyal gràfica

En l'estudi de consciència de percepció mútua entre usuaris en un entorn virtual [2] i en un altre estudi de predicció de col·lisions i la prevenció d'aquestes [3] es proposa, respectivament, mostrar un senyal de stop o directament emblanquir tota la pantalla i mostrar el senyal de stop. En els dos estudis es mostra que a velocitats reduïdes els usuaris són capaços d'evitar gran part de les col·lisions.

### 4.4. Visibilitat dels avatars

Aquesta tècnica es basa en la instanciació d'avatars que representen la posició i el comportament de cada usuari en l'escena virtual. Els avatars més elaborats tenen una major taxa d'èxit en interaccions socials [4]. En l'estudi es comprova que depenent de l'objectiu que els usuaris hi han de complir, el disseny dels avatars influeixen en el compliment d'aquest objectiu. En el cas d'escenaris interactius i orientats a completar objectius es prefereix avatars més abstractes i amb menys detall mentre que els avatars més realistes són preferibles en escenaris d'exploració [5].

Donats els estudis anteriors s'ha decidit pel nostre projecte usar com a detecció i percepció dels altres usuaris la instanciació d'avatars els quals reflectiran la posició i l'estat de les seves accions a través de les animacions corresponents. En els nostres experiments, complementant els estudis ja practicats, estudiarem el comportament dels dos usuaris més enllà de les col·lisions, observant el posicionament relatiu entre els dos agents depenent de l'acció que han de desenvolupar en cada moment i determinar quins són els factors que creen accions que no coincideixen amb un comportament humà natural en un entorn real.

## 5. Abast

En els experiments d'aquest projecte es treballarà amb dos usuaris simultàniament. Per tenir una prèvia orientació primer s'haurà d'investigar els estudis ja realitzats i identificar quins són els punts crítics a l'hora de tractar amb el comportament humà dins d'un entorn de realitat virtual. Com que els experiments es duran a terme en un entorn virtual reduït (per exemple, una habitació virtual) tots els experiments seran dirigits de manera que els resultats siguin complets i específics en aquest context, i per tant, les conclusions dels nostres resultats, encara que puguin servir per entorns on els espais d'interacció siguin més extensos, se centraran només a obtenir resultats per espais reduïts.

El hardware amb què treballarem serà el HTC VIVE Headset, que incorpora un visor individual i dos comandaments que serviran per interaccionar amb el món de forma directa. Aquests aparells tenen cert marge d'error quant a captació de moviment però, per aquest projecte, l'error el considerarem nul, ja que no és suficientment significatiu perquè afecti l'experiència de l'usuari.

### 5.1. Possibles obstacles

En els següents apartats es descriuran els possibles obstacles que ens podem trobar en el transcurs del projecte.

### 5.2. Programació de realitat virtual

Tot i que tinc experiència amb l'eina Unity 3D [6], no en tinc programant entorns de realitat virtual. Un dels possibles obstacles pot ser treballar amb l'eina de desenvolupament Steam VR, eina que proporciona els pluguins necessaris per connectar tot el hardware necessari (per exemple, casc i comandaments). Pot ser un problema perquè no només és una tecnologia amb la qual jo no he

treballat anteriorment sinó que, al ser uns components relativament nous, a la xarxa hi ha molt poca informació i poc contrastada.

### 5.3. Accessibilitat

Donat que a casa meva no dispo de hardware necessari per treballar amb entorns de realitat virtual, m'he de desplaçar fins al Centre de realitat virtual (CRV) de la UPC on tinc assignats uns horaris específics. Aquest fet suposa poca flexibilitat a l'hora de treballar i es pot donar que en algun moment no pugui dedicar tot el temps que hi sigui necessari.

## 6. Metodologia i rigor

### 6.1. Eines de desenvolupament

En el desenvolupament d'aquest projecte es treballarà amb Unity 3D (<https://unity3d.com/>) que és un motor de videojocs amb el que es crearà l'entorn virtual. Com a entrades i sortides d'àudio, moviment i localització usarem el HTC Headset juntaments amb dos comandaments que simularàn les mans. Per la creació i controlador de connexions amb un servidor al núvol usarem un *framework* anomenat Photon (<https://www.photonengine.com/en/PUN>) . Aquest *framework* proporciona poder crear, a través de Unity 3D, un servidor al núvol. Photon de forma automàtica administra les connexions als seus servidors i connexions entre usuaris.

### 6.2. Eines de seguiment

Per tal de seguir un ordre i poder preveure el temps que ens ha d'ocupar cada fase del projecte usarem la metodologia de treball Scrum[7] el qual es basa en el compliment de certes fites que els propis desenvolupadors o directors marquen dins d'un marge de temps a respectar. En aquest cas usarem una pàgina web que es diu Scrumme ([www.scrumme.com](http://www.scrumme.com)) on es crea un *backlog* que es una llista on s'anoten tots els objectius que s'han de complir per acabar el projecte i d'aquesta llista, setmanalment (aproximadament), es creen *sprints* que són un subgrup d'aquests objectius, que s'hauran d'assolir en el temps estipulat.

### 6.3. Mètodes de validació

Per tal que hi hagi un seguiment i jo com a desenvolupador pugui anar comprovant que la feina que faig està sent la ideal pel projecte i els resultats esperats, es farà un seguiment a través del

meu tutor cada dues setmanes. D'aquesta manera l'Alejandro revisarà els objectius que he realitzat i valorarà el seu resultat per poder assignar-me nous objectius o haver de refer uns altres.

## 7. Planificació

### 7.1 Descripció de les tasques

En aquest apartat s'explicaran el seguit de tasques que s'han de realitzar per tal que el projecte sigui finalitzat i compleixi les restriccions de temps degudament. Les tasques que s'especificaran són les que actualment ja han estat definides, però és possible que al llarg del transcurs del projecte certes fites siguin afegides, eliminades o modificades.

En el final de l'apartat es troba el diagrama de Gantt de tot el projecte. La data de començament del projecte és 16 de febrer de 2018 i la finalització el 28 de maig de 2018.

Les tasques es separaran en grups que definiran la fase del projecte on s'han de dur a terme. Aquests grups són:

- Definició del projecte.
- Preparació de l'entorn.
- Desenvolupament de l'escena.
- Experimentació amb usuaris i recoll·lecció de dades.
- Tractament de dades.
- Redacció de la memòria, l'annex i la documentació.

#### 7.1.1. Definició del projecte (duració aproximada: 7 dies)

Fase on, juntament amb el director, es dissenya i es concreta l'abast del projecte. Prèviament s'ha d'obtenir documentació sobre l'estat de l'art relacionat i treure'n conclusions. Els recursos necessaris durant aquesta fase seran els articles que el director del projecte proporcionarà.

#### 7.1.2. Preparació de l'entorn (duració aproximada 14 dies)

Fase on s'ha preparar tot el hardware i software amb el que es treballarà. S'ha de provar, instal·lar en cas necessari, detectar error i fer proves per comprovar que funciona tot correctament.

Els recursos necessaris durant aquesta fase seràn la documentació online que es troba a les plataformes web corresponents.

### 7.1.3. Desenvolupament de l'escena (duració aproximada: 28 dies)

En aquesta fase s'ha de desenvolupar l'escena que s'utilitzarà en els experiments. Aquesta fase és una de les més complexes (i per tant, més extensa), ja que l'escena que s'ha d'implementar requereix un grau alt de programació. Aproximadament, en la tercera setmana de desenvolupament d'aquesta secció hi haurà una reunió entre el desenvolupador (jo) i el director per comprovar l'estat de desenvolupament i afegir o modificar certs aspectes que s'hi hagin de corregir.

Els recursos necessaris durant aquesta fase seran tota la dcumentació que hi ha a la xarxa sobre Unity i sobre el Steam VR i sobre el hardware HTC Vive. També es consultarà informació a la xarxa sobre la programació i immersió en contextos virtuals.

### 7.1.4. Experimentació amb usuaris i recol·lecció de dades (duració aproximada: 14 dies)

Fase on es duren a terme experiments utilitzant l'escena prèviament desenvolupada. Durant aquesta fase se seleccionaran subjectes que seran sotmesos als nostres entorns virtuals i es recol·lectaran totes les dades necessàries per treure'n conclusions. El factor que més pot dificultar aquesta fase és el fet que faci falta gent externa al projecte, ja que pot suposar un problema de coordinació d'horari.

Els recursos necessaris durant aquesta fase seran els propis subjectes que col·laboraràn en els nostres experiments.

### 7.1.5. Tractament de dades (duració aproximada: 14 dies)

En aquesta fase s'ha de treure conclusions de les dades obtingudes dels experiments.  
Recursos necessaris encara no definits.

### 7.1.6. Redacció de la memòria, l'annex i la documentació (duració aproximada: 21 dies)

S'haurà de redactar tot el contingut de la memòria que serà entregada. Per dur a terme aquesta fase farà falta recollir de tots els procediments efectuats així com les dades obtingudes i les seves conclusions.

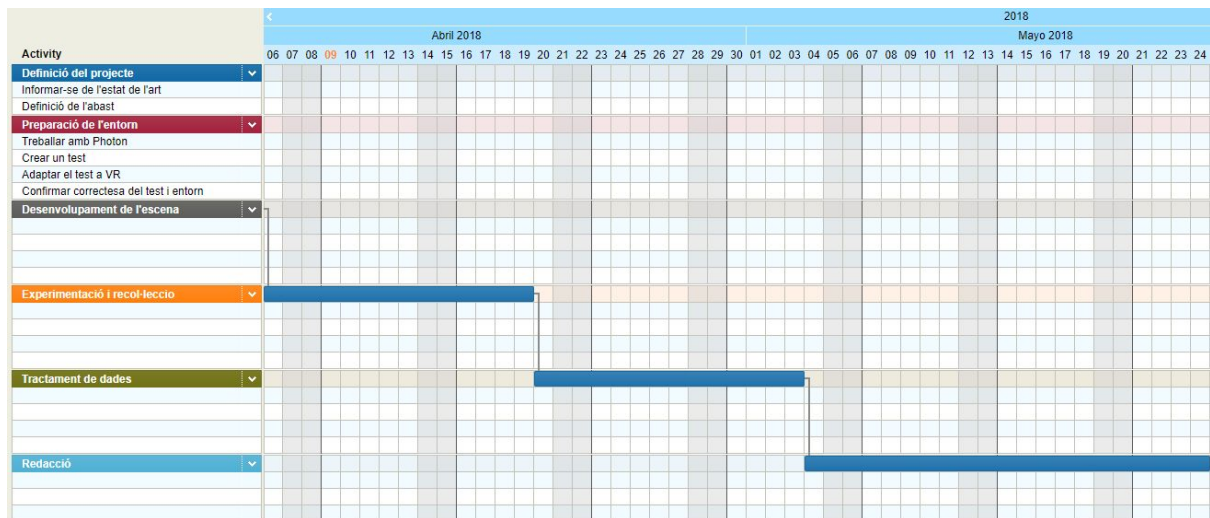
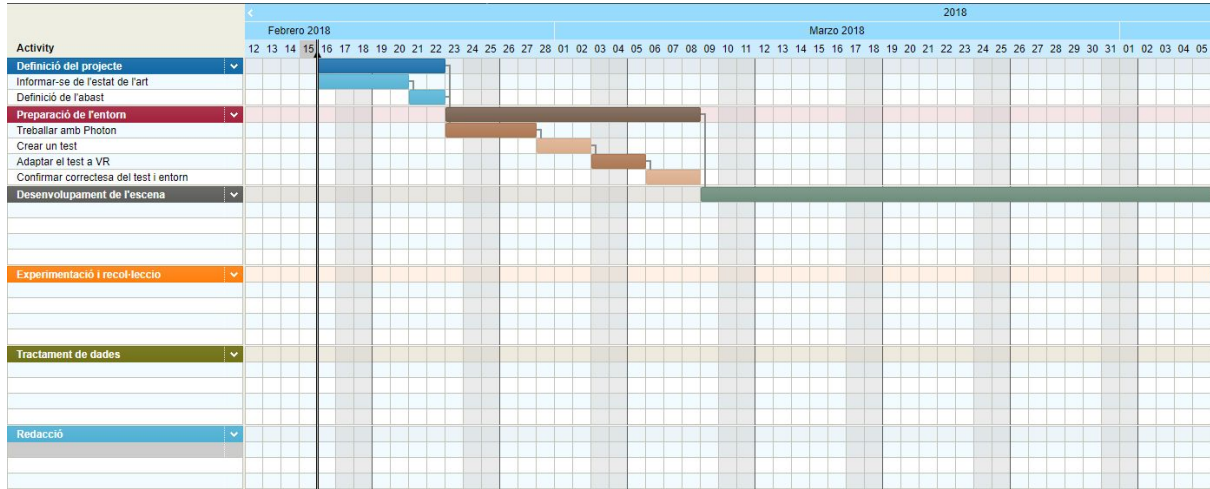
## 8. Duració total

Fase	Duració (dies)
Definició del projecte	7
Preparació de l'entorn	14
Desenvolupament de l'escena	28
Experimentació i recollida de dades	14
Tractament de dades	14
Redacció	21
Total	98

## 9. Valoració d'alternatives

Les alternatives que sorgeixin al llarg del desenvolupament del projecte es valoraran en reunions junt amb el director i es contemplarà de fer una petita implementació que ocupi el menor temps possible però que ens pugui guiar a l'hora de decidir si seguir o no amb l'actual decisió. Usarem la metodologia Scrum (a través de l'eina online Scrumme) que ens permetrà afegir certa tasca o modificar una anterior i mantenir un seguiment dels pre-requisits a l'hora de començar una següent tasca.

## 10. Diagrama de Gantt





## 11. Autoavaluació del domini actual de la competència de sostenibilitat

Després de realitzar l'enquesta d'autoavaluació sobre sostenibilitat [8] m'he adonat que no estic gens informat sobre com treballar seguint mètodes sostenibles. En primer lloc pregunta sobre la coneixença de la literatura relacionada amb la problemàtica sobre les causes, conseqüències i solucions respecte a la problemàtica social i econòmica. Actualment desconec gairebé totalment sobre aquests temes. Aquesta resposta serà la tònica general de l'avaluació: desconeixença total en temes com resolució de problemes relacionats amb l'anàlisi de la sostenibilitat d'un projecte; sóc incapaç d'identificar les causes d'un problema relacionat amb les TIC i possibles conseqüències; gairebé incapaç de subministrar noves idees o solucions en un projecte per fer-lo més sostenible.

L'enquesta també pregunta sobre la capacitat de mesurar l'impacte ambiental usant els indicadors adequats. No conec els indicadors i crec que això és important, ja que l'impacte que causi un projecte pot ser molt important per a després corregir-lo i fer-lo més sostenible si és possible. Per poder fer aquesta millora en sostenibilitat i proposar solucions s'ha de conèixer les tecnologies "sostenibilistes" aplicables a un projecte TIC, les quals desconec, juntament amb totes les problemàtiques relacionades amb l'accessibilitat, l'ergonomia i la seguretat dels productes i projectes TIC així com la problemàtica associada a la justícia total, igualtat, i diversitat i transparència.

Les preguntes següents sí que he pogut contestar de forma més afirmativa, preguntes com si tracto de maximitzar l'impacte positiu de la meva activitat professional sobre la societat o si els meus projectes intenten contribuir a millorar el bé comú de la societat. Penso que la feina que intento fer pot tenir pes a l'hora de millorar la vida de la gent i de forma indirecta millorar la sostenibilitat del planeta.

En conclusió, m'he adonat que em falta aprendre sobre sostenibilitat en desenvolupament de projectes TIC i tot i que sí que sóc conscient que és necessari, he d'invertir temps en aquest aprenentatge.

## 12. Dimensió Econòmica de la matriu de Sostenibilitat: Pressupost

En aquest apartat es detallaran tots els costos que suposa dur a terme aquest projecte. Aquests costos seran dividits en quatre grups: recursos humans, hardware, software i els costos indirectes. Per obtenir un resultat el més fidel possible a la realitat usarem els temps de desenvolupament que s'han especificat en l'entrega anterior (Lliurament 2).

### 12.1. Pressupost de recursos humans

Per al desenvolupament d'aquest projecte, en la majoria de fases l'únic agent que farà falta és un desenvolupador. Aquest desenvolupador s'encarrega de definir el projecte, desenvolupar-lo, provar-lo i corregir els seus errors, recollir les dades i treure'n les conclusions. Només en la fase d'experimentació amb usuaris farà falta agents externs.

Rol	Unitats	(€)/hora	Hores	Cost(€)
Desenvolupador	1	50	560	28.000
Tester	4	10	6	240
Total			566	28.240

### 12.2. Pressupost de hardware

En aquest apartat es tindrà en compte el maquinari necessari per desenvolupar l'experiment així com posar-lo en pràctica. S'ha considerat que amb un ordinador de gamma mitjana alta es pot dur a terme el projecte tot i que actualment s'està efectuant amb dos ordinadors simultàniament.

Producte	Unitats	Cost(€)	Vida útil	Amortització(€)
Ordinador	1	1.000	5 anys	66.66
HTC Vive	2	700	4 anys	116.66
Total		2.400		183

## 12.3. Pressupost de software

En aquest apartat es detallaran els costos del software necessari per al projecte.

Producte	Unitats	(€)	Vida útil	Amortització
Unity 3D	1	140*	-	140
Windows 10	1	120	4 anys	10
Total		260		150

\*Unity 3D es paga mensualment, en aquest cas s'ha calculat el cost de 4 mesos que és el que dura el projecte.

## 12.4.Despeses indirectes

Despeses derivades del consum d'electricitat, material d'oficina i reparacions (aproximació).

Producte	Preu(€)	Unitats	Cost(€)
Electricitat	0.12€/kWh	70.000 kWh	8.400
Oficina	200	-	240
Reparacions	200	2	400
Total			9.040

## 12.5. Control de gestió

La gestió serà portada per l'únic desenvolupador seguint les pautes estipulades en aquest document. Com que existeix la possibilitat d'imprevists com avaries, en el pressupost ja s'ha afegit cert cost de reparacions més un marge de contingència (aplicat a la taula de Cost total) per poder cobrir costos extres no previstos.

## 12.6. Cost total

Pressupost amb el cost total pel desenvolupament del projecte. Tot el pressupost s'ha fet considerant uns valors aproximats per tant el resultat podria no ser exacte.

Concepte	Cost
Recursos humans	28.240
Hardware	183
Software	153
Indirectes	9.040
Contingències	10%
Total	41.377

## 13. Dimensió Econòmica de la matriu de Sostenibilitat: Reflexió

- Respecte el Projecte posat en producció (PPP): Reflexió sobre el cost que has estimat per la realització del projecte?

Considero que es un cost ajustat i competitiu, ja que tots els costos que s'han considerat són mínims per la realització del projecte.

- Respecte la Vida Útil: Com es resolen actualment els aspectes de costos del problema que vols abordar (estat de l'art)?

Actualment els aspectes de costos es resolen amb finançament, que pot ser aconseguint un o més inversors que cobreixin part o totes les despeses o sinó finançant amb un banc.

- En què millorarà econòmicament (costos...) la teva solució respecte a les existents?

En que quant a recursos humans fa falta molta menys gent que en els altres projectes. De fet el mínim i suficient és una persona que duu a terme la majoria de la feina.

## 14. Dimensió Ambiental de la matriu de Sostenibilitat

- Respecte el Projecte posat en producció (PPP): ¿Has estimat l'impacte ambiental que tindrà la realització del projecte? No.

- Respecte el Projecte posat en producció (PPP): ¿T'has plantejat minimitzar-ne l'impacte, per exemple, reutilitzant recursos?

Sí. És possible comprar material de segona mà en el cas dels cascs HTC o alguns components dels ordinadors. Després aquest material es pot revendre i així ser més sostenibles.

- Respecte la Vida Útil: Com es resol actualment el problema que vols abordar (estat de l'art)?, i. ¿En què millorarà ambientalment la teva solució respecte a les existents?

Com s'ha dit en la pregunta anterior, part del material que adquirim pot ser reutilitzat i de la mateixa manera un cop finalitzat el projecte aquest pot ser venut perquè sigui reutilitzat.

## 15. Dimensió Social de la matriu de Sostenibilitat

- Respecte el Projecte posat en producció (PPP): Què creus que t'aportarà a nivell personal la realització d'aquest projecte?

Experiència i coneixement. És molt el que estic aprenent investigant com per exemple el funcionament de la realitat virtual. També estic adquirint coneixements de comportament humà i les reaccions bàsiques d'aquests donats certs estímuls. També estic aprenent com planejar i dur a terme un projecte d'aquestes dimensions.

- Respecte la Vida Útil: Com es resol actualment el problema que vols abordar (estat de l'art)? i. ¿En què millorarà socialment (qualitat de vida) la teva solució respecte les existents?

Els nostres resultats podran usar-se per millorar la immersió dins d'un entorn col·laboratiu virtual, d'aquesta manera fent que la socialització dins d'aquest entorn sigui molt més satisfactòria.

- Respecte la Vida Útil: Existeix una necessitat real del projecte?

No és una necessitat primària però sí que es un pas important a l'hora de seguir investigant en la immersió de la realitat virtual.

## 16. Desenvolupament

### 16.1. Introducció

En aquest apartat s'explicarà tots els temes tractats en el desenvolupament d'aquest projecte, així com les decisions preses i les modificacions que s'han anat implementant pel correcte desenvolupament d'aquest.

Aquest apartat se subdividirà en quatre apartats bàsics que seran els següents:

1. **Configuració de l'entorn.** Configuració de Unity, incloent-hi la configuració de l'entorn online amb l'asset Photon Network, disseny i implementació de l'habitació virtual, els menús i finalment la definició i configuració i interacció de l'entorn del hardware HTC Vive.
2. **Configuració de l'experiment.** Disseny de l'experiment, comportament dels avatars en l'entorn virtual i comportament dels puzzles que s'han de completar i la interacció de l'usuari amb aquests.
3. **Recol·lecció i visualitzador de dades.** Obtenció, guardat i visualització de les dades dels usuaris.
4. **Execució de l'experiment.** Apartat on s'explicarà com s'ha efectuat l'experiment.

### 16.2. Configuració de l'entorn

En aquest apartat s'explicarà com s'ha implementat tot l'entorn virtual del nostre projecte amb l'eina Unity 3D i els corresponents assets complementaris.

### 12.2.1. Creació de l'habitació en Unity

L'habitació virtual s'ha creat en Unity a través d'elements que aquesta mateixa eina ens proporciona de manera gratuïta. Aquests elements són bases primàries les quals s'han modificat perquè donin forma a l'habitació dissenyada.

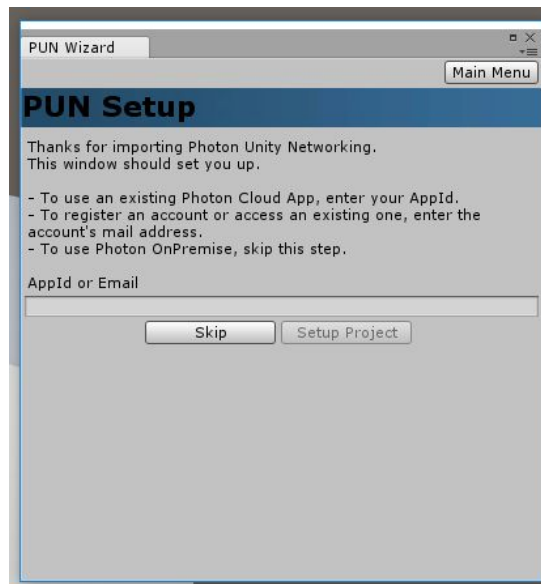
- **Terrain.** Aquest element és el que actuarà de terra on se sostindran tots els elements de l'habitació. S'ha creat un terrain de 25x25 on s'ha afegit un material amb la textura que ens ha semblat adient.
- **Plane.** Són elements que actuaran com a parets de l'habitació, aquests són quads que als quals se'ls ha assignat una textura. (Tota textura ha estat mipmapejada per millorar el rendiment a l'hora de renderitzar l'escena). També s'ha afegit un plane a cada peça del trencaclosques, on s'ha assignat la textura que li correspon depenent del trencaclosques corresponent.
- **Cube.** Element en forma de cub que s'ha usat per crear les dues taules, llenços, les dues plataformes on s'ubiquen les peces del trencaclosques i les mateixes peces del trencaclosques.
- **Light.** Element que afegeix una font de llum amb intensitat modificable a l'escena. Aquesta pot ser definida com:
  - Point: Punt de llum que s'expandeix en totes les direccions.
  - Spot: Punt de llum que emet els seus rajos dins d'un con modificable.
  - Directional light: Font de llum present en tota l'escena de la qual pots modificar la seva direcció.

### 12.2.2. Entorn online

Per la configuració online s'ha usat un asset gratuït de Unity anomenat Photon Network. Aquest asset ens proporciona un servidor en el núvol juntament amb totes les comandes necessàries per crear una habitació on connectar cada client i tots els mètodes de comunicació entre aquests.



Per identificar el servidor s'ha d'introduir l'adreça de la teva aplicació proporcionada en la pàgina web de Photon (Imatge 1).



*Imatge 1: Finestra on introduir l'adreça del servidor.*

Un cop identificat, per connectar-se al servidor, s'han usat dues comandes que connecten l'aplicació al servidor de Photon i un cop connectat sol·licites crear o unir-te a la sala que hagi definit (Imatge 2).

```
void Start () {
    PhotonNetwork.ConnectUsingSettings(version);
}

void OnConnectedToMaster()
{
    PhotonNetwork.JoinOrCreateRoom("sala1", new RoomOptions { MaxPlayers = 10 }, null);
}
```

*Imatge 2: Comandes per connectar l'usuari al servidor.*

### 12.2.3. Menús

L'aplicació que s'ha desenvolupat està formada per dos modes diferents, un d'ells és l'escena on es realitzaran els experiments i l'altre mode és el visualitzador, escena on es poden visualitzar gràficament les dades recollides en els experiments. Per poder seleccionar entre aquests dos modes, en iniciar l'aplicació apareix un menú on fer la selecció de mode (Imatge 3).

Per aconseguir persistència de les dades entre càrregues de menús s'ha creat un element anomenat Memòria el qual crida la funció DontDestroyOnLoad() que defineix l'objecte com a un element que no s'ha de destruir entre càrregues. Aquest objecte Memòria conserva les dades de l'avatar seleccionat per l'usuari, el mode amb el qual s'està treballant actualment i la ubicació del centre de l'escena.



*Imatge 3: Menú inicial*

Si es tria la primera opció, Experiment VR, es presentarà un segon menú (Imatge 4) on es podrà triar entre els dos models d'avatar disponibles juntament amb el mode d'execució desitjat. Aquests modes fan referència als 3 nivells de sincronització entre l'animació i els passos de l'avatar i l'usuari amb què es vol condicionar l'entorn amb el qual treballaràn els usuaris (tot i que aquest mode es podrà canviar en temps d'execució).



*Imatge 4: Menú de selecció de d'avatar*

#### 12.2.4. Definició de l'entorn del hardware HTC Vive

El hardware que s'ha usat per submergir l'usuari en l'entorn de realitat virtual ha estat el HTC Vive format per diversos components:

- **Headset.** Casc que incorpora una pantalla de 2880 x 1600 píxels de resolució amb una refresh rate de 90 Hz. Aquest casc té sensors per detectar la seva rotació i velocitat. En el nostre experiment s'ha utilitzat un casc amb cable i un casc sense fils.
- **Controllers.** Controladors els quals els usuaris porten a les seves mans. Disposen de sensors que registren la seva pròpia rotació i velocitat. En el nostre experiment només usarem el de la mà dreta i serà necessari per interactuar amb la nostra escena.
- **Tracker.** Dispositiu que va annexat a qualsevol objecte o part del cos el qual es vulgui tenir constància de la seva posició i rotació dins de l'entorn virtual. En aquest cas, aquest tracker va col·locat al peu esquerre i ens servirà per determinar amb quin peu s'ha començat a caminar.
- **Base station.** Dispositius que envien un senyal electromagnètic per detectar la posició relativa a ells mateixos dels elements anteriors. Són necessaris dos base stations, situats un a cada vèrtex de l'habitació, per poder obtenir les dades correctament.

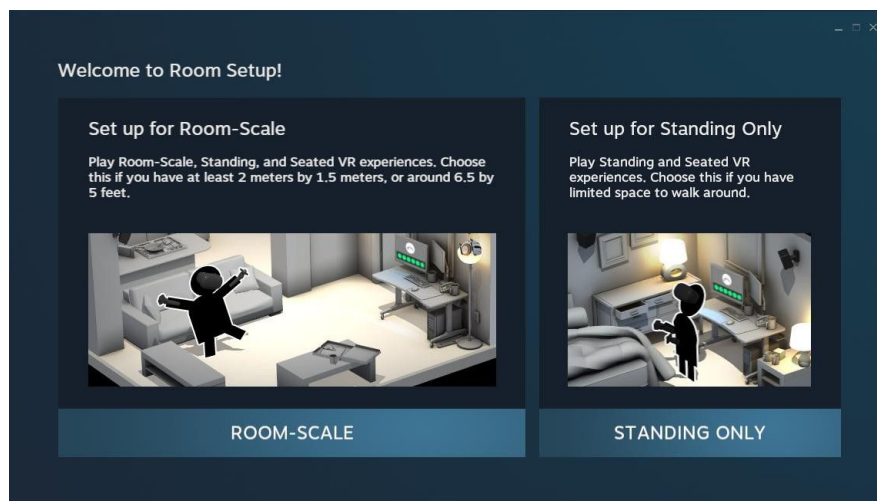


#### 12.2.5. Configuració de l'entorn del hardware HTC Vive

##### 12.2.5.1. Steam VR

Per poder utilitzar el hardware de HTC Vive en l'entorn de Unity, s'ha usat els plugins de Steam VR que ens proporcionen un conjunt de biblioteques necessàries per poder emparellar els dispositius amb l'aplicació.

En un primer pas, perquè el sistema operatiu es comuniqui amb els dispositius de HTC, s'ha de configurar l'aplicació d'escriptori Steam VR (Imatge 5). S'ha de configurar de manera que es calibri, l'altura del terra, l'orientació de la sala i la zona jugable (espai real on es defineix zona segura per on l'usuari es pot moure sense col·lidir amb obstacles).



*Imatge 5: Aplicació per calibrar l'entorn virtual*

La incorporació de Steam VR en el projecte de Unity es fa de manera que s'ha d'importar certs elements dins de l'escena que s'està desenvolupant. Aquests elements són els següents:

- **Camera Rig.** Element que inclou l'script el qual localitza i intercanvia informació amb els controladors (base stations, controllers i trackers). També inclou un script que dibuixa en l'escena la zona que l'usuari ha definit com a zona jugable.
- **Steam VR.** Element que incorpora l'script Steam VR Render, que s'encarrega de renderitzar totes les càmeres de Steam VR.

Per defecte, els índexs que s'assignen a cada dispositiu de l'escena és assignat de manera automàtica en quant Steam VR detecta aquests dispositius. En la meua aplicació, per poder tenir sempre els mateixos índexs assignats a cada tipus de dispositiu (controlador, tracker o base station), s'ha creat un script que s'encarrega d'administrar els dispositius de manera que detecta de quin tipus són i els assigna els ID en funció d'aquest tipus.

### 12.2.5.2. VRTK - Virtual Reality Toolkit

Com que el desenvolupador, en molts casos no disposava de dos cascs de HTC per poder comprovar el correcte funcionament de l'aplicació, s'ha usat el simulador de realitat virtual de la llibreria gratuïta VRTK. Aquesta llibreria aporta un administrador de dispositius de realitat virtual que, en temps d'execució, activa els pluguins necessaris per fer funcionar el hardware connectat en aquell moment. En el cas en què no hi hagi cap dispositiu de realitat virtual connectat, VRTK proporciona un simulador que es pot manejar fàcilment amb el teclat i el ratolí.

En l'imatge 6 es veu com el VRTK Manager incorpora diverses opcions de hardware de realitat virtual més un simulador.



*Imatge 6: Components principals en la jerarquia de Unity 3D*

A part del simulador, VRTK proporciona diferents scripts que han facilitat la interacció dels controladors amb l'entorn de l'experiment. Aquests scripts són els següents:

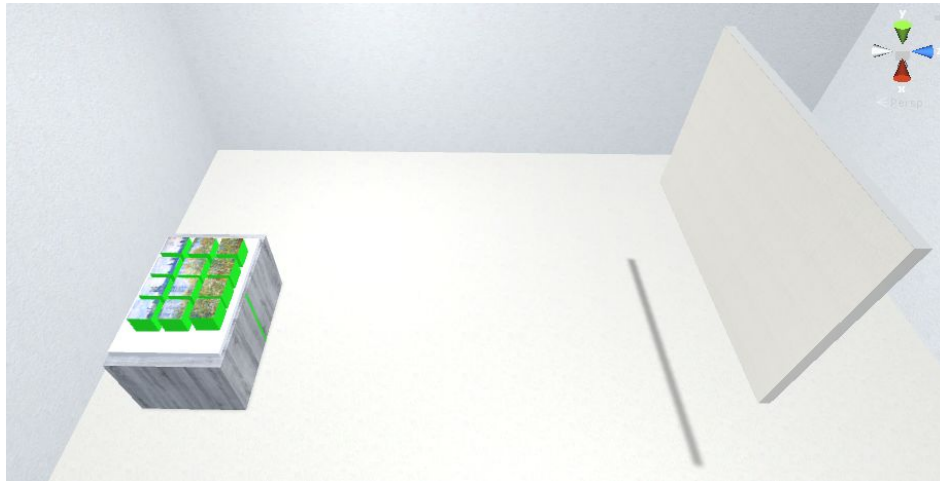
- **VRTK Interact touch/ Interact Grab.** A aquests dos scripts van assignats al controlador que es vol usar per agafar els objectes de l'escena. Proporcionen poder tocar i/o agafar aquests objectes i poder transportar-los per l'escena.
- **VRTK Interactable Object.** Script que s'assigna a cada una dels elements que vols que puguin ser interaccionats pel teu controlador. En el nostre cas, aquest script, ha estat assignat a cadascuna de les peces dels dos trencaclosques, ja que aquestes han de ser transportades per l'usuari.

## 16.3. Configuració de l'experiment

En aquest apartat s'explicarà com s'ha dissenyat i implementat l'experiment en Unity 3D. També s'explicarà com funcionen els avatars i quines interaccions tenen els usuaris amb l'entorn 3D.

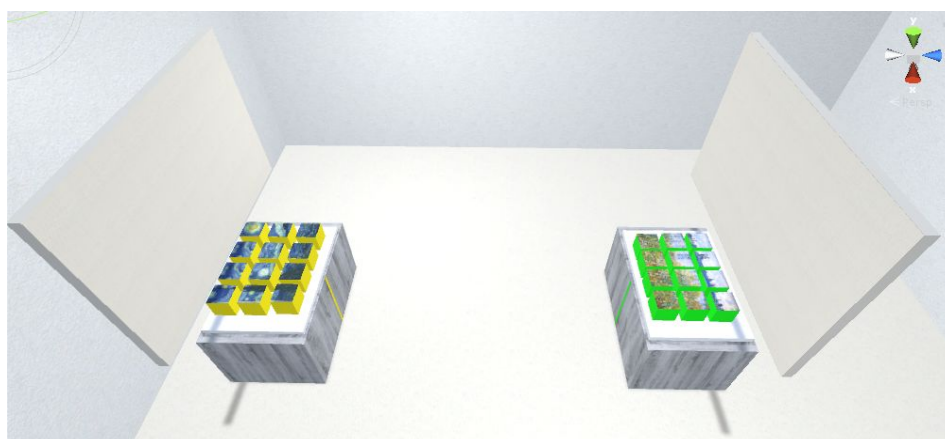
### 16.3.1. Disseny de l'habitació

Per la creació de l'habitació virtual principal on es duran a terme els experiments, s'ha agafat com a referència els models que s'han dissenyat sobre paper junt amb els directors del projecte. Aquests models en un principi incorporaven una taula amb les peces d'un trencaclosques que s'havien d'anar col·locant en un llenç a l'altra punta de l'habitació (Imatge 7).



*Imatge 7: Primera implementació de l'habitació*

Com que el comportament que es volia forçar entre els dos usuaris era un encreuament, es va decidir modificar l'habitació afegint una segona taula juntament amb un segon llenç de manera que els usuaris s'haguessin de creuar per cada peça que es volgués col·locar (Imatge 8).



*Imatge 8: Implementació final de l'habitació*

### 16.3.2. Configuració dels avatars

Pel nostre experiment, s'ha decidit que en l'escena s'instanciaran els avatars que representaran als usuaris dins de l'escena virtual de manera que els dos usuaris es poden veure a ells mateixos i als demés. Hi haurà dos avatars disponibles per triar.

Perquè l'avatar representi a l'usuari el màxim fidel possible, s'ha decidit els següents comportaments.

#### 16.3.2.1. Posicionament dels avatars

L'avatar seguirà la posició virtual de la càmera que representa el cap de l'usuari. Aquest mantindrà la seva posició vertical mentre que seguirà la rotació en l'eix Y de l'usuari.

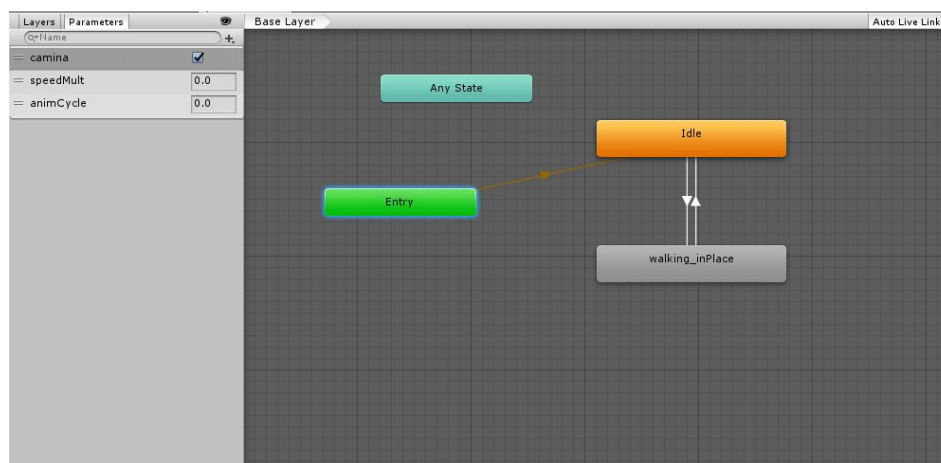
#### 16.3.2.2. Escalat dels avatars

Per una millor immersió dels usuaris, és necessari que els avatars siguin escalats respecte la mida real de la persona. S'ha programat una funció que quan es pitja una tecla del teclat (A) l'avatar s'escala automàticament de manera que el seu cap coincideixi amb l'alçada del casc de HTC dins de l'escena virtual. En qualsevol moment es poden usar les fletxes verticals del teclat per ajustar manualment l'escalat de l'avatar.

#### 16.3.2.3. Animació dels avatars

Els avatars tindran dues animacions disponibles que són caminar i idle (Imatge 9). L'animació per defecte serà idle que serà la que s'executi mentre l'usuari està quiet en el lloc. L'animació de caminar s'accionarà en quant el headset detecti un mínim de velocitat en els eixos X i Z. La velocitat de l'animació es modificarà, a través d'un paràmetre, de forma directament relativa a la velocitat del casc, de manera que la velocitat en què l'avatar camina sigui el més fidel a la realitat.

Per determinar amb quin peu l'usuari comença a caminar, i accionar l'animació en conseqüència, en el peu esquerre de l'usuari s'ubica un tracker el qual, instants després que l'usuari comença a caminar, envia la seva posició per saber quin peu té més avançat respecte a la posició del cap. Un cop determinat el peu amb el que s'ha començat, a través d'un altre paràmetre, s'acciona l'animació en el frame desitjat.



*Imatge 9: Diagrama d'estats de les animacions de l'avatar*

### 16.3.3. Configuració dels trencaclosques

En el nostre experiment cada usuari ha de completar un trencaclosques de manera individual. Aquests dos trencaclosques estan compostos per 12 peces els quals formen una imatge d'un quadre conegut. Aquestes peces estan ubicades en la seva taula d'inici, respectivament, i han de ser transportades una per una per l'usuari fins al llenç on seran col·locades.

Pel correcte funcionament d'aquest trencaclosques s'han hagut d'implementar els següents apartats.

#### 16.3.3.1. Cos no kinematic

Un cos kinemàtic és aquell que no és afectat per forces externes (com la gravetat). En el nostre cas, com que es desitjava que l'entorn sigui el més fidel a la realitat possible, en un principi es van fer proves marcant les peces com a no kinemàtiques, però va sorgir un problema: Des de l'usuari que observava, de manera molt ràpida es veia com les peces vibraven entre la posició de la mà i una posició llunyana a la posició anterior.

El que causava aquest comportament era que, entre actualitzacions del servidor, l'observador aplicava a la peça que veia en l'altre usuari la força de la gravetat, causant que aquesta caigui fins que el servidor actualitzava la posició real de la peça (la mà de l'usuari) i la transportava un altre cop fins a la posició anterior. Per solucionar aquest error es va crear un mètode que modificava el component kinemàtic de manera que quan una peça era agafada, en tots els clients, aquesta peça era modificada com a kinemàtica (no és afectada per la gravetat) i en quant aquesta era deixada a la taula o en l'aire, perdia aquesta propietat.



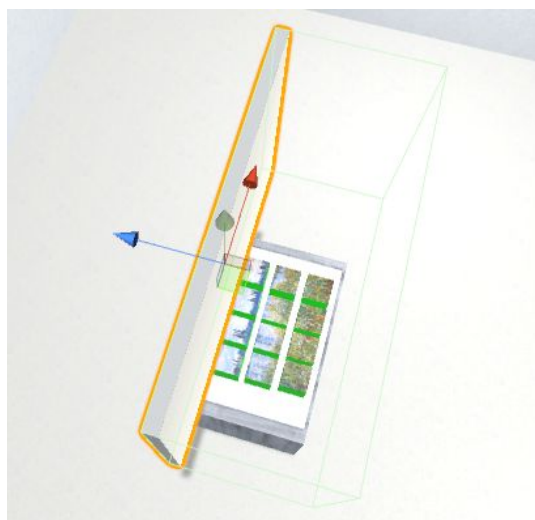
### 16.3.3.2. Reset del trencaclosques

S'ha habilitat un mètode que retorna les peces al seu punt inicial (recuperant les seves propietats kinemàtiques originals). Aquest mètode és accionat en dues situacions específiques: 1) La peça ha estat deixada anar per l'usuari que la tenia agafada i fa més de tres segons que es troba en aquest estat. Aquesta acció només restableix la posició de la peça afectada. 2) S'ha accionat la tecla R del teclat, acció que fa que totes les peces en l'escena tornin al seu punt inicial.

Com que el punt de repòs de la peça a sobre de la taula pot variar, ja que és possible moure la posició de tota l'habitació, sempre que es fa un reset de la peça o del trencaclosques sencer es comprova el vector de desplaçament que ha intervingut respecte a l'anterior punt i se li suma a la posició de repòs anterior.

### 16.3.3.3. Col·locar peça

Primerament, en l'objecte del llenç s'ha definit una hitbox que funcionarà com a trigger amb la peça. És a dir, quan la peça entri dins d'aquesta zona (línies verdes en la imatge 10), el paràmetre de la peça que defineix si aquesta està en la zona correcta per ser col·locada, serà canviat a cert, de manera que quan l'usuari la deixi anar, aquesta serà col·locada a la posició que li pertoca dins del llenç.



*Imatge 10: En línies verdes, la hitbox del llenç*

Quan l'usuari deixa anar la peça i aquesta està dins de la hitbox del llenç que li pertoca, es crida la funció Col·locar. Aquesta crida calcula la posició que li pertoca a la peça en funció de l'identificador que inicialment se li ha assignat, de manera que automàticament és col·locada en el lloc que li correspon dins de l'imatge total del quadre.

Per calcular la posició on col·locar la peça dins del llenç es té en compte la cantonada esquerra superior del llenç més un offset en X i en Y que es calcula multiplicant el número de ID per la suma de la mida de la peça i el forat que es volgués deixar entre peces. En un principi la posició en l'eix Z es calculava a partir de la posició del llenç més un offset en aquest mateix eix depenent de quin quadre era en concret, però aquesta configuració donava alguns errors quan hi havia desplaçament de l'escena. La solució va ser definir la posició en l'eix Z respecte a un objecte buit situat davant de cada llenç, d'aquesta manera no s'havia de modificar el valor Z manualment.

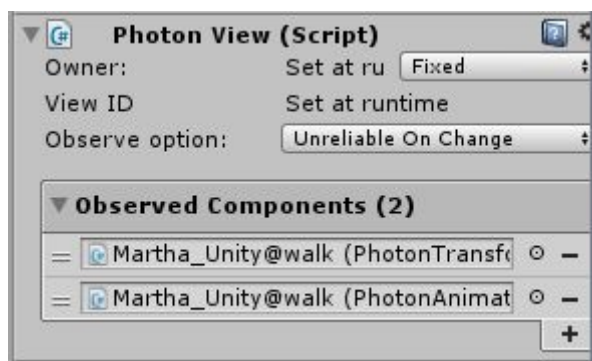
#### 16.3.4. Configuració de l'interacció en xarxa

Com que la nostra aplicació està connectada a un servidor en línia la manera de modificar els elements en la nostra escena és diferent a com ho fariem en una aplicació no online.

##### 16.3.4.1. Configuració de l'avatar en la xarxa

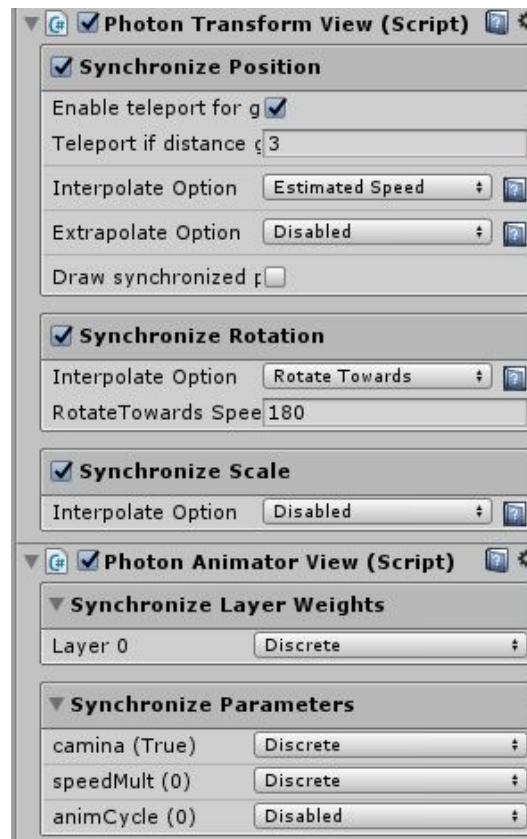
Per instanciar l'avatar usarem una comanda especial que ens proporciona Photon Network, anomenada *Instantiate*, que instancia l'objecte que se li indica en l'escena i manté actualitzada la informació que se li indica amb el servidor.

La manera d'indicar quins components es volen mantenir actualitzats amb el servidor és adjuntant, en primer lloc, un script a l'objecte anomenat *Photon View*. Aquest script s'encarrega de mantenir actualitzada l'informació que se li especifica (Imatge 11).



*Imatge 11: Component Photon View assignat a un avatar*

Aquesta informació que volem actualitzar, en el nostre cas, serà la posició i rotació de l'avatar juntament amb la seva animació, que obtindrem adjuntant dos scripts més a l'objecte: *Photon Animation View* i *Photon Transform View* (Imatge 12).



*Imatge 12: Components Photon Transform view i Photon Animation View assignats a l'avatar*

#### 16.3.4.2. Funcions RPC (Remote Procedure Call)

Les funcions RPC són funcions que es criden no només en un executable sinó en tots els altres clients simultàniament. És a dir, que són funcions que es criden quan es desitja que un comportament succeeixi en tots els usuaris en el mateix instant de temps. Per poder utilitzar aquestes funcions és necessari tenir les aplicacions connectades en xarxa.

En el nostre projecte s'han definit un seguit de funcions RPC per sincronitzar certes accions entre els dos usuaris:

- **Reset de la peça.** És necessari que quan es resetja qualsevol peça o el trencaclosques sencer s'efectuï en els dos usuaris, ja que els dos estan veient la mateixa escena.
- **SetRB.** Aquesta funció s'encarrega d'alterar les propietats kinemàtiques de les peces. Per exemple, quan un usuari agafa una peça s'executa aquesta funció que modifica les

propietats de la peça per cada usuari, de manera que en els dos clients la peça no és afectada per la gravetat.

- **StardRecord.** Funció que s'encarrega de indicar a l'aplicació de cada client que ha de començar a guardar l'informació de l'usuari.
- **Save.** Funció que crea, per cada client, un document on es guarda tota la informació capturada fins aquest moment.
- **MourePeça.** S'ha vist que la posició dels avatars és actualitzada de manera automàtica per la llibreria de Photon, però en el cas de les peces va sorgir un problema: les peces havien de ser instanciades en l'inici de l'execució igual que els avatars. Aquesta restricció ens impedia col·locar les peces manualment des de l'editor, disminuint la flexibilitat de l'experiment a l'hora de provar diferents disposicions del trencaclosques.

La solució va ser programar la sincronització de la posició de les peces de manera manual amb una funció RPC. Aquesta funció es crida des del client que està movent la peça amb el controlador i es reproduïx en els altres clients que actualitzen la posició de la seva peça amb els paràmetres que el primer a cridar la funció ha passat. Per compensar la latència que hi ha entre crides de funció i la latència amb el servidor, la posició que s'actualitza es calcula amb una interpolació entre la posició anterior i l'actualitzada.

Aquesta funció també està assignada a un element buit que serveix per sincronitzar el moviment de l'inverse kinemàtics de la mà dels avatars.

#### 16.3.4. Posicionament de l'escena

En una aplicació en realitat virtual, per defecte, el punt central (0,0,0) de l'aplicació se situarà al centre de l'espai jugable definit per l'usuari, i el centre d'aquest espai està definit segons la posició relativa a les base stations de HTC. Això vol dir que si un usuari, en el seu PC, defineix la zona jugable en una posició relativa a les bases concreta, i l'altre usuari, per error, defineix aquesta zona en un altre punt diferent, el resultat serà que la posició de l'escena no coincidirà pels dos usuaris en el mateix espai físic. Per solucionar aquest error, s'ha programat una funció que localitza la posició de les base stations dins de l'espai virtual i mou l'escena fent coincidir el centre de l'escena amb el centre de la diagonal de les base stations. Amb la tecla P es pot alternar entre aquesta posició o la definida per defecte.

## 16.4. Recol·lecció i visualitzador de dades

En aquest apartat es defineix quins sistemes s'han usat a l'hora d'actualitzar, guardar, carregar i visualitzar les dades enregistrades, de l'usuari.

### 16.4.1. Recol·lecció de dades

Per poder guardar les dades, s'ha creat una llista d'Instants. Aquests Instants són instàncies d'un struct el qual té com a variables la posició, orientació, booleà que indica si es porta peça o no, l'instant de temps en què s'ha pres la instància i finalment la posició dels peus de l'avatar. La distància en temps entre cada instant guardat, actualment, és de 0.5 segons.

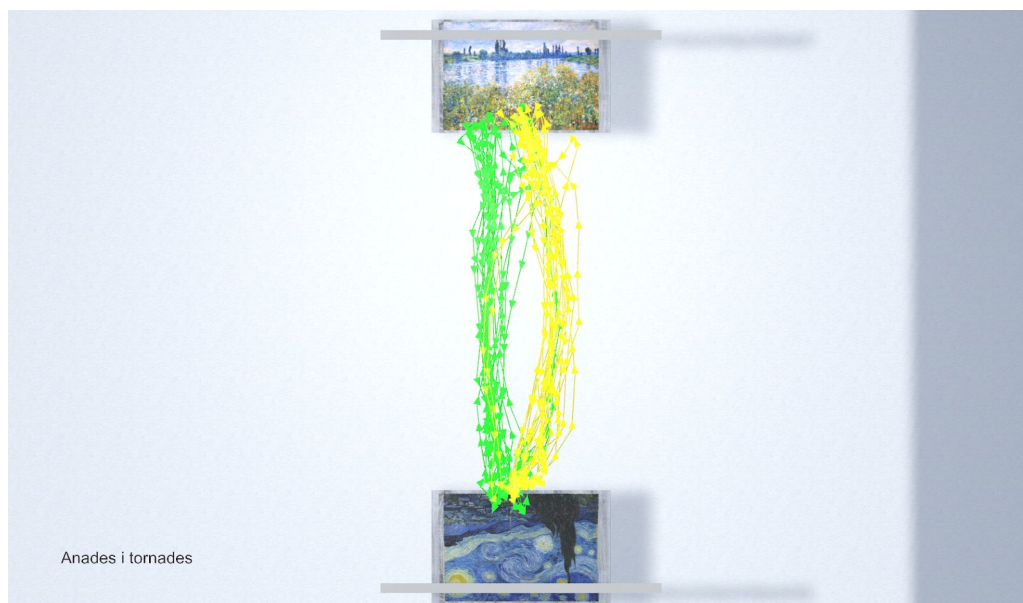
La posició de l'avatar sempre es defineix com la posició relativa al centre actual de l'escena, restant el vector de desplaçament del centre actual respecte al (0,0,0), de manera que tant si es decideix centrar l'escena al centre de les base station o del contrari es vol conservar el centre definit per defecte sempre en el visualitzador les dades es veurà que comparteixen espai.

Com s'ha explicat, aquestes dades, cada 0.5 segons es van guardant en Instants en una llista fins que l'experiment s'acaba. És en aquest moment en què es crida la funció RPC Save que s'encarrega de, simultàniament en els PCs dels dos usuaris, crear un document de text on per cada fila s'escriurà un Instant convertit en text. En l'arxiu, per millorar la seva comprensió, la funció Save defineix el nom d'aquest amb la data actual més l'usuari corresponent (1 o 2). També inclou una capçalera on es defineix una llegenda de les dades escrites més un enter que representa el mode en el qual s'ha desenvolupat l'experiment.

### 16.4.2. Visualitzador de dades

L'aplicació disposa d'un visualitzador que pot ser executat en el primer menú que es mostra en iniciar l'aplicació. Un cop s'ha seleccionat, s'obre un explorador d'arxius amb la carpeta Data\_recolectada oberta on s'haurà de seleccionar l'arxiu que es vol visualitzar. Un cop seleccionat, el programa obre els dos arxius amb el mateix nom (menys el número d'usuari), llegeix tots els instants escrits, assigna un color, verd o groc, a cada document i crea, per cada instant de temps llegit, una instància en forma de triangle en l'escena representant la posició i orientació d'aquell instant (imatge 13). Depenent de si es porta peça o no, el color serà fosc o clar respectivament.

En temps d'execució, utilitzant les tecles 1,2 i 3, es pot alternar entre visualitzar totes les dades, només les anades (desplaçaments amb peça) o només les tornades (desplaçaments sense peça). També, amb la barra d'espai, es pot rotar la càmera 180 graus respecte l'eix Y de l'escena per orientar l'escena al revés. Amb la tecla O s'obre un nou explorador d'arxius per carregar unes noves dades.



*Imatge 13: Visualització del posicionament de dos usuaris en el temps*

## 16.5. Realització de l'experiment

En aquest apartat s'explicarà com s'ha efectuat l'experiment del dia 29 de maig del 2018.

### 16.5.1. Preparació

L'experiment es va realitzar a quatre parelles d'alumnes que actualment estan cursant l'assignatura IDI en la facultat d'Informàtica de Barcelona de la UPC. Les seves edats estan entre els 20 i 26 anys.

Cada parella va realitzar quatre intents, on cada intent era un muntatge complet del trencaclosques, aplicant en cada intent un mode dels tres modes de sincronització. El quart mode era el mode sense casc, on simplement s'havien de situar el casc a sobre del cap i realitzar l'experiment en l'entorn real amb caixes de cartró.



Prèviament se'ls va facilitar un document on s'explicava breument l'experiment que estaven a punt d'efectuar, juntament amb els seus riscos. En aquest document explicatiu s'explicava sobretot el hardware que utilitzarien, però no en cap cas el comportament que havien de tenir respecte a les tasques que s'anaven a trobar, ja que l'experiment havia de ser el màxim natural possible.

Tan bon punt cada membre de la parella es va col·locar el seu casc, se'ls va fer situar un a cada punta de l'habitació virtual, davant de les seves respectives taules. Un cop situats se'ls va explicar la tasca que havien de realitzar amb les peces del trencaclosques, mentre se'ls va convidar a agafar una peça de prova i intentar col·locar-la al seu lloc corresponent.

### 16.5.2. Execució

Un cop els dos usuaris havien confirmat que havien entès la tasca a realitzar es va donar la marca auditiva que els indicava l'inici de l'intent. L'experiment continuava fins que els dos trencaclosques eren completats i s'indicava als usuaris que es podien treure els cascs.

Cal indicar que el dia de l'experiment només disposàvem d'un casc inalàmbic, per tant creava un escenari on un dels usuaris podia caminar lliurement per la sala però l'altre usuari portava arrossegant el cable. Es va intentar compensar el condicionament del cable amb el posicionament d'una persona encarregada de recollir-lo i adequar-lo al moviment de l'usuari perquè aquest no notés cap diferència respecte l'inalàmbic (Imatge 14).



*Imatge 14: Usuaris duent a terme l'experiment. El cable es recollia per no dificultar el moviment*

### 16.5.3. Treball posterior

Un cop els usuaris acabaven cada intent, se'ls feia treure els cascs i omplir un formulari on se'ls preguntava sobre la previsió del comportament de l'altre usuari. En total omplien tres tests, un per cada mode de sincronització. Pel mode sense casc no es va omplir cap formulari.

## 17. Conclusions

Un cop finalitzat el projecte s'ha pogut completar dos dels tres objectius que s'havien marcat en l'inici.

En el que respecta a l'experiment que s'havia dissenyat, s'ha pogut completar en la seva totalitat tot i que certes restriccions s'han hagut de resoldre amb la tecnologia de la qual s'ha disposat:

- La primera restricció ha estat només disposar d'un tracker per usuari, cosa que ens ha limitat a l'hora de sincronitzar l'animació de l'avatar amb el peu amb el qual començava a caminar l'usuari. Els possibles errors d'animació que hagi pogut causar aquest inconvenient és possible que hagi afectat en els usuaris a l'hora d'interaccionar i per tant els resultats poden no ser el més fidels possibles a la realitat.
- En segon cas, donat que disposàvem només d'un casc inalàmbic considerem que el comportament dels usuaris eren afectats a l'hora de triar el camí per creuar l'habitació. Aquest comportament l'hem vist confirmat quan, els usuaris en l'intent on es posaven els cascs al cap (no als ulls), veien on s'ubicava el cable i corregien la seva trajectòria per no coincidir-hi.

Quant a recollida i visualització de dades s'ha pogut realitzar en la seva totalitat, creant un sistema que registra, guarda i visualitza les dades sòlidament i amb totes les dades necessàries per poder fer un estudi extens i significatiu.

Tenint en compte això, treu la conclusió que els resultats són molt positius. S'ha creat un experiment útil al qual se li pot fer moltes modificacions per crear altres experiments i poder crear un conjunt d'estudi que aportï dades significatives en el camp del comportament dels usuaris a l'hora d'esquivar i interaccionar amb obstacles en un entorn de realitat virtual.



## 18. Treball futur

Els resultats d'aquest projecte són satisfactoris però proposem certes tasques a fer per poder millorar i completar el nostre projecte en un futur.

En primer lloc la realització del nostre últim objectiu que no s'ha pogut completar, que és l'estudi de les dades recollides en els experiments realitzats. Tot i que es poden realitzar més experiments amb més usuaris, de moment amb les dades que es van treure el dia 29 de maig de 2018 en el centre de realitat virtual de la UPC es pot fer un estudi d'on es treguin les primeres conclusions.

En segon lloc, com s'ha comentat en les conclusions, s'haurà d'ampliar el nombre de trackers per millorar el seguiment dels peus i així millorar la sincronització entre els usuaris i els seus avatars. Aquesta millora pot influir en el nivell d'immersió dels usuaris i així aconseguir unes dades més fiables.

En tercer lloc es proposa fer els experiments amb dos cascs inalàbrics per poder proporcionar als usuaris un entorn sense condicionaments de moviment.

Finalment, a partir d'aquestes tres millores, es pot considerar ampliar l'experiment incorporant diferents tasques a realitzar, més enllà de muntar un trencaclosques, de manera que els usuaris puguin realitzar tasques menys tedioses i no se'ls faci tan repetitives.

## Referències

- [1] K. Blom and S. Beckhaus. Virtual travel collisions: Response method influences perceived realism of virtual environments. *ACM Trans. Appl. Percept.*, 10(4):25:1– 25:19, 2013.
- [2] I. Podkosova and H. Kaufmann. Mutual Proximity Awareness in Immersive MultiUser Virtual Environments with Real Walking. In *ICAT-EGVE 2015*
- [3] J. Holm. Collision prediction and prevention in a simultaneous multi-user immersive virtual environment. Master's thesis, Miami University, 2012.
- [4] D. Banakou, K. Chorianopoulos, and K. Anagnostou. Avatars' appearance and social behavior in online virtual worlds. In *PCI 2009*, pages 207–211, 2009.
- [5] [20] J.-L. Lugin, J. Latt, and M. Latoschik. Anthropomorphism and illusion of virtual body ownership. In *ICAT-EGVE 2015*.
- [6] [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) *Unity is a cross-platform game engine developed by Unity Technologies,<sup>[2]</sup> which is primarily used to develop both three-dimensional and two-dimensional video games.*
- [7] [https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)) *Scrum is an agile framework for managing work with an emphasis on software development. It is designed for teams of three to nine developers who break their work into actions that can be completed within timeboxed iterations.*
- [8] Cuestionario de Estudiantes de Ingeniería Informática.  
<https://docs.google.com/forms/d/e/1FAIpQLSeIZixKIUFbCn1oVkd2JM3yxCc208E85RgKZclKd8eUu3GvBg/viewform>

## Apèndix

En aquest apartat s'inclouen les parts més rellevants del codi. Tot el codi font pot ser localitzat en el corresponent directori

### Posició, rotació i escalat de l'avatar

```
// si el model es el meu que es mogui amb el meu headset
if (photonView.isMine)
{
    // moviment de l'avatar modo 3d person
    float currentAngle = transform.eulerAngles.y;
    float desiredAngle = target.transform.eulerAngles.y;

    Quaternion rotation = Quaternion.Euler(0, desiredAngle, 0);
    transform.position = target.transform.position - (rotation * offset);
    transform.LookAt(target.transform);
    transform.rotation = Quaternion.Euler(0f, transform.rotation.eulerAngles.y, 0f);
    transform.position = new Vector3(transform.position.x, 0f, transform.position.z);

    float scaleFactor = 0.02f;
    float instaScaleFactor = 0.002f;
    float marge = 0.025f;

    if (Input.GetKeyDown(KeyCode.A))
    {
        while (headAvatar.transform.position.y < headset.transform.position.y - marge)
        {
            transform.localScale += new Vector3(scaleFactor, scaleFactor, scaleFactor);
        }
        while (headAvatar.transform.position.y > headset.transform.position.y + marge)
        {
            transform.localScale -= new Vector3(scaleFactor, scaleFactor, scaleFactor);
        }
    }

    if (Input.GetKey(KeyCode.UpArrow))
    {
        transform.localScale += new Vector3(instascaleFactor, instascaleFactor, instascaleFactor);
    }
    else if (Input.GetKey(KeyCode.DownArrow))
    {
        transform.localScale -= new Vector3(instascaleFactor, instascaleFactor, instascaleFactor);
    }
}
}
```

## Animació de l'avatar

```

headVel = new Vector3(cam.velocity.x, 0, cam.velocity.z);

if (photonView.isMine && caminant && anim.GetBool("camina")) anim.SetFloat("speedMult",
    Mathf.Min(headVel.sqrMagnitude * modificador, 1.3f));
// Debug.DrawRay(headset.transform.position, capDireccioV);

if (!caminant && !calculantPeu && photonView.isMine && headVel.sqrMagnitude >= 0.05f)
{
    calculantPeu = true;
    lastMostraT = Time.time; // - offsetMostres / numMostres;
}

if (calculantPeu && photonView.isMine && Time.time - lastMostraT > 0.3) //&& mostresPeu.Count
{
    calculantPeu = false;
    caminant = true;

    Debug.Log("animo");
    Vector3 peuHeadV = peu.transform.position - headset.transform.position;
    Vector3 capDireccioV = Quaternion.Euler(0f, headset.transform.eulerAngles.y, 0f) *
        Vector3.forward;
    bool davantEsq = Vector3.Dot(capDireccioV.normalized, peuHeadV.normalized) >= 0f;
    anim.SetBool("camina", true);
    if (davantEsq) anim.SetFloat("animCycle", 0f + 0.17f);
    else anim.SetFloat("animCycle", 0.5f + 0.17f);
}

if (caminant && photonView.isMine && headVel.sqrMagnitude < 0.007f)
{
    anim.SetBool("camina", false);
    anim.SetFloat("animCycle", 0f);
    caminant = false;
}

```

## Col·locació de les peces

```
[PunRPC]
void colocar()
{ // colocar la pesa per tots els clients

    colocarA.Play();
    rb.isKinematic = true;
    transform.localScale += new Vector3(scaleFactor, scaleFactor, scaleFactor);

    gap = 0.0f;
    cubeSize = (0.95f) * scaleFactor;

    if (artista == "VAN") lienzo = GameObject.Find("Lienzo Van");
    else if (artista == "MONET") lienzo = GameObject.Find("Lienzo Monet");
    else Debug.LogError("artista de la pesa no especificat");

    int fila = id / 4;
    int col = id % 4;

    if (artista == "VAN")
    {
        ini = lienzo.transform.position + new Vector3(- cubeSize * 1.5f, cubeSize, 0f);
        ini = new Vector3(ini.x, ini.y, lienzo.transform.Find("iniPesa").transform.position.z);

        transform.rotation = Quaternion.identity;
        transform.Rotate(new Vector3(90, 180, 0), Space.World);

        transform.position = ini + new Vector3(col * cubeSize, -fila * cubeSize, 0);
    }
    else if (artista == "MONET")
    {
        ini = lienzo.transform.position + new Vector3(cubeSize * 1.5f, cubeSize, 0f);
        ini = new Vector3(ini.x, ini.y, lienzo.transform.Find("iniPesa").transform.position.z);

        transform.rotation = Quaternion.identity;
        transform.Rotate(new Vector3(90, 0, 0), Space.World);

        transform.position = ini + new Vector3(-col * cubeSize, -fila * cubeSize, 0);
    }
    colocat = true;
}
```

## Captura i guardat de l'informació del usuari

```
[PunRPC]
public void startRecord()
{
    tIni = Time.time;
    iniA.Play();
    Debug.Log("Grabant");
    posicionesCap = new List<instant>();
    grabant = true;
}

void Update () {

    if (Input.GetKeyDown(KeyCode.I) && !grabant) // alterna el mode entre grabar i no grabar
    {
        GetComponent<PhotonView>().RPC("startRecord", PhotonTargets.All);
    }
    else if ((Input.GetKeyDown(KeyCode.S)) && grabant) // apretar S per crear fitxer i guardar les dades
    {
        // li passo la hora en la que apreto perque ens els dos pc tingui el mateix nom
        GetComponent<PhotonView>().RPC("save", PhotonTargets.All, System.DateTime.Now.ToString("MM-dd-yy_hh-mm-ss-"));
    }

    if (headset == null) headset = GameObject.Find("Camera (eye)");
    else if (grabant && Time.time - lastCapture > frameTime)
    {
        instant newInstant = new instant(headset.transform.position - mem.centre, headset.transform.rotation.eulerAngles,
            portaPesa, (Time.time-tIni).ToString(), peuDret.position - mem.centre, peuEsq.position - mem.centre);
        posicionesCap.Add(newInstant);
        lastCapture = Time.time;
    }
}
```



```
[PunRPC]
public void save(string name) // guarda en el teu pc l'arxiu de la teva trajectoria
{
    grabant = false;

    file = File.Open(pathT + "/" + name + user + ".txt", FileMode.OpenOrCreate, FileAccess.ReadWrite);
    StreamWriter writer = new StreamWriter(file);
    writer.WriteLine("#Mode: " + mem.mode);
    writer.WriteLine("#Punts d'inici: " + (liniaM.position - mem.centre) + ", " + (liniaV.position - mem.centre));
    writer.WriteLine("#Llegenda: headPosition|headDirection|portaPeça|peuEsquerrePos|peuDretPos|timeStamp &: rotacio");
    writer.WriteLine("&|" + background.transform.rotation.eulerAngles.x + ", " + background.transform.rotation.eulerAngles.y
        + ", " + background.transform.rotation.eulerAngles.z);

    foreach (instant inst in posicionsCap)
    {
        writer.WriteLine(
            inst.pos.x.ToString() + "," +
            inst.pos.y.ToString() + "," +
            inst.pos.z.ToString() + "|" +

            inst.direccio.x.ToString() + "," +
            inst.direccio.y.ToString() + "," +
            inst.direccio.z.ToString() + "|" +

            inst.portaPesa.ToString() + "|" +

            inst.peuE.x.ToString() + "," +
            inst.peuE.y.ToString() + "," +
            inst.peuE.z.ToString() + "|" +

            inst.peuD.x.ToString() + "," +
            inst.peuD.y.ToString() + "," +
            inst.peuD.z.ToString() + "|" +

            inst.timeStamp
        );
    }
    writer.Close();

    Debug.Log("Guardat");
}
```